

WHAT IS CLAIMED IS:

1. A floating point multiplier circuit configured for performing extended-precision multiplication of an N-bit multiplicand value by an M-bit multiplier value, wherein N and  
5 M are positive integers, said floating point multiplier circuit comprising:

partial product generation logic configured to generate a plurality of partial  
products from said multiplicand value and said multiplier value, wherein  
said plurality of partial products corresponds to a first portion of said  
10 multiplier value during a first partial product execution phase, and wherein  
said plurality of partial products further corresponds to a second portion of  
said multiplier value during a second partial product execution phase;

a plurality of carry save adders coupled to said partial product generation logic  
15 and configured to accumulate said plurality of partial products generated  
during said first partial product execution phase into a redundant product  
during a first carry save adder execution phase, and further configured to  
accumulate said plurality of partial products generated during said second  
partial product execution phase into said redundant sum during a second  
20 carry save adder execution phase; and

a first carry propagate adder coupled to said plurality of carry save adders and  
configured to reduce a first portion of said redundant product to a  
multiplicative product during a first carry propagate adder phase, and  
25 further configured to reduce a second portion of said redundant product to  
said multiplicative product during a second carry propagate adder phase;

wherein said first carry propagate adder phase begins after said second carry save adder execution phase completes.

2. The floating point multiplier circuit as recited in claim 1, wherein:

5

said plurality of carry save adders is further configured to perform an arithmetic left shift on said redundant product accumulated during said first carry save adder execution phase by a number of bits corresponding to said first portion of said multiplier value; and

10

said plurality of carry save adders is further configured to accumulate a result of said arithmetic left shift with said second portion of said plurality of partial products into said redundant product during said second carry save adder execution phase.

15

3. The floating point multiplier circuit as recited in claim 1, wherein:

said first portion of said multiplier value corresponds to a higher-order portion of said multiplier value;

20

said second portion of said multiplier value corresponds to a lower-order portion of said multiplier value;

said first portion of said redundant product corresponds to a lower-order portion of said redundant product; and

25

said second portion of said redundant product corresponds to a higher-order portion of said redundant product.

4. The floating point multiplier circuit as recited in claim 1, wherein:
- said redundant product includes a Q-bit sum term and an R-bit carry term;
- 5
- said first carry propagate adder includes a plurality of operand inputs, wherein each operand input includes at most P bits; and
- each of P, Q, and R is a positive integer, P is less than Q, and P is less than R.
- 10
5. The floating point multiplier circuit as recited in claim 1 further comprising a plurality of rounding adders coupled to said plurality of carry save adders and configured to produce a respective plurality of rounded multiplicative products.
- 15
6. The floating point multiplier circuit as recited in claim 5, wherein each rounding adder is further configured to:
- receive a respective rounding constant;
- 20
- accumulate said respective rounding constant with a first portion of said redundant product into a rounded redundant product during said first carry propagate adder phase;
- reduce a first portion of said rounded redundant product to a given respective rounded multiplicative product during said first carry propagate adder phase; and
- 25

reduce a second portion of said rounded redundant product to said given  
respective rounded multiplicative product during said second carry  
propagate adder phase.

5     7.     The floating point multiplier circuit as recited in claim 1, further configured for  
performing pipelined reduced-precision multiplication of said N-bit multiplicand value by  
an S-bit multiplier value with a single partial product execution phase, a single carry save  
adder execution phase, and a single carry propagate adder phase, wherein S is a positive  
integer and S is less than or equal to  $N/2$ , and wherein each of said single partial product  
10    execution phase, said single carry save adder execution phase, and said single propagate  
adder phase may accept a new reduced-precision multiplication operation during a given  
execution cycle.

8.     The floating point multiplier circuit as recited in claim 1, wherein said partial  
15    product generation logic includes a plurality of Booth encoders and a plurality of Booth  
multiplexers.

9.     The floating point multiplier circuit as recited in claim 1, wherein M is equal to  
2Y, wherein said first portion of said multiplier value includes the most significant Y bits  
20    of said multiplier value, and wherein said second portion of said multiplier value includes  
the least significant Y bits of said multiplier value.

10.    A method, comprising:  
  
25       receiving an N-bit multiplicand value and an M-bit multiplier value;  
  
generating a plurality of partial products from said multiplicand value and said  
multiplier value, wherein said plurality of partial products corresponds to a

first portion of said multiplier value during a first partial product execution phase, and wherein said plurality of partial products further corresponds to a second portion of said multiplier value during a second partial product execution phase;

5

accumulating said plurality of partial products generated during said first partial product execution phase into a redundant product during a first carry save adder execution phase;

10

accumulating said plurality of partial products generated during said second partial product execution phase into said redundant product during a second carry save adder execution phase;

15

reducing a first portion of said redundant product to a multiplicative product during a first carry propagate adder phase; and

reducing a second portion of said redundant product to said multiplicative product during a second carry propagate adder phase;

20

wherein said first carry propagate adder phase begins after said second carry save adder execution phase completes.

11. The method as recited in claim 10, further comprising :

25

performing an arithmetic left shift on said redundant product accumulated during said first carry save adder execution phase by a number of bits corresponding to said first portion of said multiplier value; and

accumulating a result of said arithmetic left shift with said second portion of said plurality of partial products into said redundant product during said second carry save adder execution phase.

5     12.     The method as recited in claim 10, wherein:

said first portion of said multiplier value corresponds to a higher-order portion of said multiplier value;

10           said second portion of said multiplier value corresponds to a lower-order portion of said multiplier value;

said first portion of said redundant product corresponds to a lower-order portion of said redundant product; and

15           said second portion of said redundant product corresponds to a higher-order portion of said redundant product.

20     13.     The method as recited in claim 10, wherein:

said redundant product includes a Q-bit sum term and an R-bit carry term;

each of said first and second portion of said redundant product includes at most P bits; and

25           each of P, Q, and R is a positive integer, P is less than Q, and P is less than R.

14. The method as recited in claim 10, further comprising:

receiving a plurality of rounding constants;

5 accumulating each rounding constant with a first portion of said redundant product into a respective rounded redundant product during said first carry propagate adder phase;

10 reducing a first portion of each said respective rounded redundant product to a given respective rounded multiplicative product during said first carry propagate adder phase; and

15 reducing a second portion of each said respective rounded redundant product to said given respective rounded multiplicative product during said second carry propagate adder phase.

15. The method as recited in claim 10, further comprising selectively performing pipelined reduced-precision multiplication of said N-bit multiplicand value by an S-bit multiplier value with a single partial product execution phase, a single carry save adder execution phase, and a single carry propagate adder phase, wherein S is a positive integer and S is less than or equal to  $N/2$ , and wherein each of said single partial product execution phase, said single carry save adder execution phase, and said single propagate adder phase may accept a new reduced-precision multiplication operation during a given execution cycle.

25

16. A microprocessor comprising:

dispatch logic configured to issue multiply instructions to a floating-point unit;  
and

5

a floating-point unit coupled to said dispatch logic and configured to:

receive an N-bit multiplicand value and an M-bit multiplier value;

10

generate a plurality of partial products from said multiplicand value and  
said multiplier value, wherein said plurality of partial products  
corresponds to a first portion of said multiplier value during a first  
partial product execution phase, and wherein said plurality of  
partial products further corresponds to a second portion of said  
multiplier value during a second partial product execution phase;

15

accumulate said plurality of partial products generated during said first  
partial product execution phase into a redundant product during a  
first carry save adder execution phase;

20

accumulate said plurality of partial products generated during said second  
partial product execution phase into said redundant product during  
a second carry save adder execution phase;

25

reduce a first portion of said redundant product to a multiplicative product  
during a first carry propagate adder phase; and



reduce a second portion of said redundant product to said multiplicative product during a second carry propagate adder phase;

wherein said first carry propagate adder phase begins after said second carry save adder execution phase completes.

17. The microprocessor as recited in claim 16, wherein:

said plurality of carry save adders is further configured to perform an arithmetic left shift on said redundant product accumulated during said first carry save adder execution phase by a number of bits corresponding to said first portion of said multiplier value; and

said plurality of carry save adders is further configured to accumulate a result of said arithmetic left shift with said second portion of said plurality of partial products into said redundant product during said second carry save adder execution phase.

18. The microprocessor as recited in claim 16, wherein:

said first portion of said multiplier value corresponds to a higher-order portion of said multiplier value;

said second portion of said multiplier value corresponds to a lower-order portion of said multiplier value;

said first portion of said redundant product corresponds to a lower-order portion of said redundant product; and

said second portion of said redundant product corresponds to a higher-order portion of said redundant product.

5     19.     The microprocessor as recited in claim 16, wherein:

said redundant product includes a Q-bit sum term and an R-bit carry term;

10            said first carry propagate adder includes a plurality of operand inputs, wherein  
              each operand input includes at most P bits; and

each of P, Q, and R is a positive integer, P is less than Q, and P is less than R.

15     20.     The microprocessor as recited in claim 16 further comprising a plurality of  
              rounding adders coupled to said plurality of carry save adders and configured to produce a  
              respective plurality of rounded multiplicative products.

20     21.     The microprocessor as recited in claim 20, wherein each rounding adder is further  
              configured to:

receive a respective rounding constant;

25            accumulate said respective rounding constant with a first portion of said  
              redundant product into a rounded redundant product during said first carry  
              propagate adder phase;

reduce a first portion of said rounded redundant product to a given respective rounded multiplicative product during said first carry propagate adder phase; and

5           reduce a second portion of said rounded redundant product to said given respective rounded multiplicative product during said second carry propagate adder phase.

22.       The microprocessor as recited in claim 16, further configured for performing  
10       pipelined reduced-precision multiplication of said N-bit multiplicand value by an S-bit multiplier value with a single partial product execution phase, a single carry save adder execution phase, and a single carry propagate adder phase, wherein S is a positive integer and S is less than or equal to  $N/2$ , and wherein each of said single partial product execution phase, said single carry save adder execution phase, and said single propagate  
15       adder phase may accept a new reduced-precision multiplication operation during a given execution cycle.

23.       A method comprising:

20       receiving an extended-precision floating-point arithmetic operation, wherein said arithmetic operation is performed in a plurality of iterations of a set of one or more floating-point operations including at least one floating-point multiplication;

25       determining an arithmetic precision generated during a given iteration of said arithmetic operation;

performing a reduced-precision multiplication operation during said given iteration of said arithmetic operation if said arithmetic precision generated during said given iteration is less than or equal to a precision of said reduced-precision multiplication operation; and

5

performing an extended-precision multiplication operation during said given iteration of said arithmetic operation if said arithmetic precision generated during said given iteration is greater than a precision of said reduced-precision multiplication operation.

10

24. The method as recited in claim 23, wherein said extended-precision floating-point arithmetic operation comprises a floating point divide operation.

25. The method as recited in claim 23, wherein said extended-precision floating-point arithmetic operation comprises a floating point transcendental operation.

15

20